2

UNIVERSITY OF MASSACHUSETTS
AT AMHERST

Natural Language Processing Laboratory
Department of Computer Science
Amherst, MA 01003

February 8, 1994

Wendy G. Lehnert
Professor of Computer Science
lehnert@cs.umass.edu
(413) 545-3639
vax: 545-1249

Ms. Shelly Young
L423 - Maryland Procurement Office
9800 Savage Road
Fort George G. Meade, MD 20755-6000

Dear Ms. Young:

This package contains Tipster Phase I deliverables from the University of Massachusetts contract MDA904-92-C-2390. We note that software deliverables are only usable via computer-readable media, so we have not included hardcopies of source code files, data files, or documentation files associated with those items. Our software deliverables can be picked up via FTP over the internet using the following instructions:

The commands to be entered by the user are underlined with "^".
Comment lines start with ">>>".

```
>>> Start up ftp.

% ftp
  ^^^

>>> Connect to the ftp host at cs.umass.edu.

ftp> open ftp.cs.umass.edu          (Internet host #: 128.119.40.244)
     ^^^^^^^^^^^^^^^^^^^^^^

>>> Enter username.

Name (ftp.cs.umass.edu:<user-id>): dodpikup
                                   ^^^^^^^^
>>> Enter password.

Password: gmpolgar
          ^^^^^^^^

>>> Show contents of "home" directory.

ftp> ls
     ^^

README
tips3.tar.Z
doc.tar.Z
autoslog.tar.Z
interfaces.tar.Z
README.tips3
```

```
>>> Retrieve top-level documentation file.

ftp> get README
   ^^^^^^^^^^

>>> Specify binary mode transfer for compressed archives.

ftp> bin
   ^^^

>>> Retrieve any or all compressed archives, e.g., tips3.tar.Z.
>>> Note that the ftp "mget" command is disabled at this ftp site.

ftp> get tips3.tar.Z
   ^^^^^^^^^^^^^^^

>>> Close connection

ftp> bye
```

If you have any questions, please let me know. We will maintain this directory for a period of 6 months, until August 15, 1994.

Sincerely,

Wendy G. Lehnert
Professor

WGL/psc
Enclosures

CC: Robert D. Powell
Monique Dillon
Director, NRL
DTIC

STATUS REPORT

Title:	Corpus-Based Knowledge Acquisition Support for Text Analysis Systems
Frequency:	Quarterly Report
Period:	October 1, 1992 to December 31, 1992
Contract No.:	MDA904-92-C-2390
Submitted by:	Wendy G. Lehnert, Principal Investigator Department of Computer Science University of Massachusetts, Amherst, MA 01002
Security Classification:	Unclassified

Distribution Statement A: Approved for Public Release; distribution is unlimited

Based on Items from the Software Development Schedule

Summary Statement: We have completed all software development tasks and milestones specific to the University of Massachusetts on time as specified in section 1.9 of our original proposal. Because of some delays associated with our subcontract to Hughes, none of the work assigned to Hughes for the first quarter has been completed.We received our 2 DECstations in December and ported our software to these machines.

A detailed breakdown of our system development effort follows.

Preliminary Design Completed at 3 months: In moving CIRCUS from the MUC terrorism domain to the TIPSTER domains, we found it necessary to rework some aspects of CIRCUS' operation in order to better accommodate fully automated dictionary acquisition. In particular, we decided to incorporate a trainable part-of-speech tagger (OTB) of our own design, as well as a new module for noun phrase analysis. These are the major system innovations that have required design efforts not outlined in our original proposal.

Acquisition of Base Lexicon from MRD Completed at 2 months: We have determined that an adequate base lexicon can be obtained using an OTB training kernel alone. No MRD is being used at this time. However, we will conduct experiments to see if an augmented dictionary drawing selected entries from an MRD is better.

Statistical Experiments Initiated after 2 months: The statistical OTB algorithm is operating at better than 90% hit rates. Higher hit rates can be obtained by augmenting the statistical algorithm with tag repair heuristics.

Port AutoSlog from MUC domain Completed at 2 months: AutoSlog has been successfully adapted to the JV domain. A few new generation patterns have been added, backpointers from JV templates to source texts are in place, and alterations to AutoSlog have been made to keep it compatible with the latest version of CIRCUS.

Assess AutoSlog Performance Initiated after 2 months: We have manually filtered a complete collection of AutoSlog concept node definitions derived from the entire JV development corpus (1000 texts). This has given us a good sense of AutoSlog's strengths and weaknesses. We are now working to improve various aspects of AutoSlog's operation in response to this manual data analysis.

Basic Click & Drag Dictionary Interface Completed after 2 months: A manual tagging interface supports OTB's training sessions. A separate interface for analysts who need to mark documents for AutoSlog is now under development.

Forms Interface for Templates Initiated after 2 months: The interface design process has begun.

Basic Drag & Drop Interface Initiated after 2 months: The interface design process has begun.

Preliminary Implementation Completed after 3 months: A newly revised CIRCUS sentence analyzer is now operating in conjunction with a part-of-speech tagger (OTB) and AutoSlog concept node definitions only. No additional dictionary support is needed. A new noun phrase analysis module has been incorporated into CIRCUS, and specialist modules have been created to create canonical representations for dates and revenue objects.

Implement Template Generators Work Delayed: Because the subcontract from the University of Massachusetts was not completed by our Office of Grants and Contracts until January, none of the first quarter work assigned to Hughes has been completed. We expect to see progress on the template generators as soon as the subcontract to Hughes is in place.

Fill 150 English JV Templates Ongoing at 3 Months: We have delivered 50 English JV templates to IDA under a December deadline as requested.

Future Plans: Our top priority is to complete the 18-month TIPSTER evaluation. Because of the subcontracting delay, it is difficult to predict a strong showing. But we expect to learn a lot from the process in any case. Our primary focus has been the JV domain, so we will spend more time working on ME during the second quarter.

STATUS REPORT

Title: Corpus-Based Knowledge Acquisition Support for Text Analysis Systems

Frequency: Quarterly Report

Period: January 1, 1993 to March 31, 1993

Contract No.: MDA904-92-C-2390

Submitted by: Wendy G. Lehnert, Principal Investigator
Department of Computer Science
University of Massachusetts, Amherst, MA 01002

Security Classification: Unclassified

Distribution Statement A: Approved for Public Release; distribution is unlimited

UMass/Hughes Tipster Project Quarterly Report

January 1 - March 31, 1993

An Assessment of the Current System

Major progress was made this quarter as a result of the 18-month evaluation and the integration of UMass sentence analysis capabilities with Hughes classifier technologies. This integration brought together 3 components designed to enhance system portability and scalability: a trainable part-of-speech tagger, a trainable concept node dictionary, and a trainable template generator.

Although we are pleased with our success in automated knowledge acquisition, our performance levels in the 18-month evaluation were admittedly disappointing. This poor showing prompted us to take a hard look at our overall system architecture. We have consequently come to the conclusion that our simple control flow from sentence analysis to template generation was too simple for the task at hand. We were expecting template generation to resolve all problems associated with discourse analysis — no other component in our architecture was responsible for discourse analysis. As a consequence, we were not meeting the challenges of co-reference and we were having difficulties identifying relational information.

There were other problems as well. Because of our self-imposed requirements for automated dictionary acquisition, our sentence analyzer was impaired by the total absence of any semantic features. We were also not handling complex noun phrases satisfactorily, having stripped out a lot of domain-dependent noun-phrase handling that was previously buried inside the CIRCUS sentence analyzer. Our commitment to automated knowledge acquisition made it clear that while we were moving in some good directions, there was still substantial work to be done.

We came out of the 18-month meeting with a better understanding of some significant problems in our current system. Additional analysis in the weeks immediately following the 18-month meeting has resulted in a revised system architecture and initial progress toward a revised implementation. As reported at our Feb. 12 site visit, one of our strongest priorities is the integration of a trainable semantic feature tagger. We are still convinced that semantic features are crucial for us but we have since come to understand that certain aspects of discourse analysis are probably at least as important as semantic features. As a result, we have reworked our overall architecture to include new capabilities that cannot be adequately addressed by sentence analysis and template generation alone.

A New and Greatly Improved System Architecture

To understand this shift in more detail, it is useful to compare the system architecture used for the 18-month evaluation, with the new one currently being implemented. Figure 1 shows the old flow of control with a strong emphasis on sentence analysis as the driving force behind template generation. Figure 2 shows how we now include key components of discourse analysis prior to template generation, along with the integration of semantic features as originally planned. Note that each new component in figure 2 represents trainable domain-specific capabilities.

In all of the previous UMass/MUC systems, concept node instantiations produced by CIRCUS were the central driving force behind all subsequent processing. We are now moving toward an architecture that is still heavily influenced by concept nodes, but equally concerned with object tracking and information extraction at the level of noun phrases as well as sentences. In our previous UMass/MUC systems, object tracking was handled by highly domain-specific code inside a memory consolidation component. In moving toward a portable technology we had to drop that component. We originally thought that the Hughes Trainable Text Skimmer (TTS-MUC3) might step in to fill that gap, but our 18-month evaluation suggests that memory consolidation and TTS-MUC3 were not very interchangeable. Now we are ready to bring back the capabilities of memory consolidation, but this time with an eye toward trainable components that acquire domain-specific knowledge during training. TTS-MUC3 did show us how to create such trainable components, but it was overly optimistic to think that one such component would suffice for all the problems associated with good memory representations.

One of the insights gained after the 18-month evaluation was an appreciation for how much work was being thrown at the template generator and how many different problems remained to be solved during template generation. We determined that the classifier technologies available to us would be more effective if we could break the template generation task down into key sub problems that could be handled independently and then reassembled for final template generation.

We have therefore arrived at a decomposition that shifts a lot of the work away from template generation per se, and addresses that same work during the phase we are now calling Discourse Analysis. *The primary job of Discourse Analysis is to produce memory tokens that represent important referents throughout the text, along with relational links between those memory tokens.* Discourse Analysis is domain-dependent insofar as a taxonomy of memory tokens is determined by the information extraction task before us, and the link-types between them are task-driven as well. But a serious effort is being made to maintain Discourse Analysis as a portable component with trainable capabilities designed to handle domain-sensitive discriminations as much as possible.

We have identified three opportunities for trainable discourse analysis: (1) appositive handling, (2) co-reference resolution, and (3) establishing relational links. *Appositive handling* refers to the problem of knowing when we have a legitimate appositive description for a single referent, as opposed to two referents that happen to be separated by a comma. Although appositives occur within the confines of a single sentence, we prefer to think of appositives as a problem in discourse analysis rather than sentence analysis because the decisions asso^iated with appositives hold more impact for discourse-level memory than sentence-level memory. *Co-reference resolution* refers to the problem of knowing when a new noun phrase refers back to a previously encountered referent. *Establishment of relational links* is needed to structure memory tokens in an associative network containing links known to be important for our information extraction requirements.

In our previous system architecture, all of these problems in Discourse Analysis were handed over to the template generator with the expectation that a single classifier might somehow be able to sort it all out. We now understand this was an unreasonably complex training task, and we need to decompose the complexity into some smaller pieces that might be more realistically handled by trainable classifiers. So the job of template generation in figure 1 is now distributed between Template Generation and Discourse Analysis in figure 2. What used to be the work of one trainable classifier is now being handed off to four separate classifiers. We believe that this is a much more realistic integration of natural language and machine learning capabilities.

Of the three targeted problems in Discourse Analysis, co-reference resolution is by far the most challenging. We have looked at this problem very carefully and believe that we have a viable strategy for handling co-reference using a trainable classifier. This key problem has assumed a central position in our new architecture, and our overall system performance will be heavily dependent on the reliability of our co-reference discriminations. Co-reference resolution is sensitive to features, semantic features, overall sentence analysis, and complex noun phrase analysis. So any propagation of errors associated with the proper identification of these features will become readily apparent as we attempt to recognize co-referent noun phrases.

Our Immediate Research Plan

Our current plan is to complete a new system implementation based on figure 2, and evaluate the effectiveness of key system components in this new architecture. This plan represents a major departure from our proposed research for the third quarter, but it seems fully justified given the results of the 18-month evaluation. We note that some of this work requires new design decisions as well as implementation efforts, so it is difficult to say how quickly we can get back on track with respect to our original plan for year 1. However, we do expect to have a stable version of the improved system running in time for the 24-month evaluation. That is a clear priority for us. More immediately, we hope to have some preliminary results on semantic feature tagging, appositive handling, co-reference resolution, relational links, and memory token generation in time for our next TIPSTER site visit which is currently scheduled for April 23.

Additional Progress

We note that additional technical progress was made during this last quarter independent of the system design issues discussed above.

- We have begun to implement an interface for text annotation that operates as a substitute for hand-coded templates to support AutoSlog dictionary construction in the absence of hand-coded templates.

- We have completed enhancements to AutoSlog in order to provide stronger dictionary coverage. In particular, we have set up a capability to generalize active/passive concept node definitions, and we have incorporated other capabilities associated with verb tense generalizations.

Publications, Presentations and Conferences Attended

- Wendy Lehnert gave an invited talk at the University of Michigan on information extraction from text (February 11, 1993)

- UMass hosted a TIPSTER site visit (February 12, 1993)

- Wendy Lehnert, Charles Dolan, and Joe McCarthy attended the TIPSTER 18-month meeting. Wendy Lehnert presented a paper on UMass/Hughes Test Results and Analysis. Charles Dolan presented a paper on the UMass/Hughes System Description. Lehnert, Dolan, and McCarthy also contributed individual workshop presentations. (February 22-24, 1993)

- Wendy Lehnert delivered a keynote address on information extraction from text at the Conference on Artificial Intelligence Applications. (March 2-5, 1993)

- Ellen Riloff presented a paper on automated dictionary construction at the Conference on Artificial Intelligence Applications. (March 2-5, 1993) "Automated Dictionary Construction for Information Extraction from Text"

- Joe McCarthy and Stephen Soderland attended the DARPA Workshop on Human Language Technologies (March 21-24, 1993)

- Ellen Riloff presented a paper at the AAAI Spring Symposium on Case-Based Reasoning and Information Retrieval (March 24-26, 1993). "Using Cases to Represent Context for Text Classification"

- Claire Cardie and Ellen Riloff attended the Sixth Annual CUNY Sentence Processing Conference at the University of Massachusetts, Amherst. Claire Cardie delivered a poster titled "Understanding Empty Category Constructions in a Semantically-Oriented Parser" (March 18-20).

- A paper by Ellen Riloff on automated dictionary construction was accepted by AAAI-93. "Automatically Constructing a Dictionary for Information Extraction Tasks".

- Two separate papers by Claire Cardie on semantic feature acquisition were accepted by AAAI-93 and Machine Learning-93. "A Case-Based Approach to Knowledge Acquisition for Domain-Specific Sentence Analysis" and "Using Decision Trees to Improve Case-Based Reasoning".

Conference Reports

Title: Corpus-Based Knowledge Acquisition Support
for Text Analysis Systems

Frequency: Quarterly Report

Period: January 1, 1993 to March 31, 1993

Contract No.: MDA904-92-C-2390

Submitted by: Wendy G. Lehnert, Principal Investigator
Department of Computer Science
University of Massachusetts, Amherst, MA 01002

Security
Classification: Unclassified

Distribution Statement A: Approved for Public Release; distribution is unlimited

CONFERENCE REPORTS

1. TIPSTER 18-month Evaluation Meeting

Date, place of conference, and attendees: February 22-24, Williamsburg, Virginia. Attendees were Charles Dolan (Hughes), Joseph McCarthy and Wendy Lehnert (UMass).

Subjects discussed: Status of TIPSTER systems. Alternative Evaluation metrics. Text processing strategies used by human analysts. Sharable resources.

The agenda of scheduled reports and discussions provided us with new insights, and we also found time to discuss difficulties we were experiencing with our UMass/Hughes system integration plan.

2. DARPA Workshop on Human Language Technology:

Date, place of conference, and attendees: March 21-24, Princeton, New Jersey. Attendees were Joseph McCarthy and Stephen Soderland (UMass).

Subjects discussed: Scheduled paper presentations. A variety of research efforts involving statistical methods and trainable NL systems. Successful integration of NL and speech technologies. Resource libraries and data collection. Application-oriented technology evaluations.

STATUS REPORT

Title:	Corpus-Based Knowledge Acquisition Support for Text Analysis Systems
Frequency:	Quarterly Report
Period:	April 1, 1993 to June 30, 1993
Contract No.:	MDA904-92-C-2390
Submitted by:	Wendy G. Lehnert, Principal Investigator Department of Computer Science University of Massachusetts, Amherst, MA 01002
Security Classification:	Unclassified

Distribution Statement A: Approved for Public Release; distribution is unlimited

UMass/Hughes TIPSTER Quarterly Report: April 1 - June 30, 1993

Project Summary

This quarter was devoted to the completion of ME and JV implementations based on the new system architecture designed during the previous quarter. New modules have been completed for these implementations and we have been able to test our newest ideas on trainable system components. A list of the new system components and capabilities follows:

- AutoSlog interface enhancements for morphology and active/passive transformations
- a location specialist for the preprocessor
- an "aka" specialist for the preprocessor
- context-free CN definitions
- CN definitions based on multiple occurrences (as opposed to one-shot instances)
- CN dictionary compression to remove functionally redundant CN definitions
- a new OTB trigram algorithm and simpler OTB tag repair module
- trainable noun phrase recognition
- trainable appositive recognition
- discourse analysis: memory token creation

In addition to the creation and integration of these new modules, we have also integrated previously existing capabilities into our new system architecture for the first time:

- the MayTag semantic tagger
- the Hughes trainable template generator[1]

During this quarter we also completed additional training and testing of specific components that support selective concept extraction with CIRCUS:

[1] The OTB part-of-speech tagger now has access to an EJV training base derived from 1009 JV sentences as well as an EME training base derived from 621 ME sentences. A stronger integration of OTB with the preprocessing specialists has also been achieved. Using a 10-fold cross validation design for training and testing, we show a 97% overall hit rate on all parts of speech for both EJV texts and EME texts. It took 16 hours to manually tag the training data used for EJV, and 10 hours to manually tag the training data for EME.

[1] Although we had TTG hooked up for the 18-month evaluation, we are now handing it memory tokens instead of CN instantiations in accordance with the new system architecture.

[2] Experiments with trainable appositive recognition indicate that we now achieve an 87% hit rate[2] on EJV appositives (true positive and true negative recognition). It took 10 hours to manually classify 2276 training instances for the appositive classifier using a training interface. We have not tested this component on EME texts but we believe the features used by this component are domain-independent.

[3] Our trainable noun phrase component decides when a noun phrase properly scopes a prepositional phrase (pp-attachment), crosses commas, or crosses conjunctions in order to pick up appositives or compound noun phrases. Experiments with trainable noun phrase recognition indicate that we can now identify EJV noun phrases perfectly 87% of the time. 7% of our noun phrases pick up spurious text (they're too long) and 6% of our noun phrases are truncated (they're too short). Our hit rates for EME are very similar: 86% for exact NP recognition, with 6% picking up spurious text and 8% being truncated. The classifier used for noun phrase recognition operates on the basis of syntactic tags and selected lexical features: no semantic feature tags are being used for noun phrase recognition (surprisingly). This means that we were able to use the exact same module for EME that was originally trained to handle EJV. The classifier was trained on 1350 EJV noun phrases examined in context. It took 14 hours to manually mark these 1350 instances using a text editor.

[4] The MayTag semantic feature tagger has been trained on 174 JV sentences containing 5591 words (3060 open class words and 2531 closed class words). Preliminary tests indicate that MayTag achieves a 74% hit rate on general semantic features (covering 14 possible tags) and a 75% hit rate on specific semantic features (covering 42 additional tags). Among other things, MayTag contributes to the recognition of company names since we do not use an explicit dictionary of company names. Our tests indicate that MayTag recognizes jv-entity (a general semantic feature tag) words with 90% recall and 77% accuracy. Interactive training for MayTag took 14 hours.

[5] Improved noun phrase recognition and enhancements to the AutoSlog interface have resulted in the largest CN dictionary to date. In EJV, AutoSlog proposed 3167 CN definitions in response to 1100 EJV templates. After manual filtering, 944 of these (30%) were retained for our CN dictionary. When we added generalizations based on active/passive transformations, verb tense variations, and singular/plural variations, the total number of CN definitions jumped to 3017. This EJV CN dictionary is 6.4 times bigger than the EJV CN dictionary used for the 18-month evaluation. In effective functionality the actual increase may be even greater because we are now "compressing" the CN dictionary to remove functionally redundant CN definitions. We had never bothered to check for redundant definitions before. The larger number of CN definitions proposed by AutoSlog has increased the amount of

[2] These results are actually based on an earlier (and smaller) training set. We have not yet completed testing for our most recently updated training set. We expect that we will see some improvement over these earlier hit rates.

time needed for manual filtering, but not too badly. It took 20 hours of manual inspection for one person to complete the EJV CN dictionary. In EME, AutoSlog proposed 2952 CN definitions in response to 1000 EME templates. After manual filtering, 2275 of these (77%) were retained for our CN dictionary. The higher retention rate for EME is due to the large number of technical keywords and fixed phrases that are useful for EME. When we added in generalizations, the number of total CN definitions went up to 4220. It took 17 hours of manual inspection to complete the EME CN dictionary.

****

With so many new trainable components and dependencies across the trainable components, the system development cycle must be carefully orchestrated so that upstream components are stabilized before training data is generated for downstream components. Our new components and their resulting dependencies now create an 8-step system development cycle (up from 3 steps at the 18-month evaluation).

Although our new system architecture entails a longer development cycle, we are very pleased with the increased effectiveness of the new design. We have successfully completed implementations for EJV and EME based on the architecture proposed during the last quarter, and we believe that this redesign has addressed the weaknesses of our 18-month system as predicted. We hope to demonstrate the validity of this claim with the results of our 24-month evaluation at the end of July. In spite of the increased overhead associated with an 8-step development cycle (as opposed to a 3-step cycle), we are still operating with a viable development cycle in terms of overall human resources and turn-around requirements. We will discuss estimated development times for porting to new domains in the next section.

We summarize our trainable system development in tables 1 and 2.

System Development Time Estimates

Our total time spent on interactive training for a single development cycle in EJV was 74 hours. Additional time was also needed to collect and process training instances from the development corpus. OTB required sentence analysis by the preprocessor. All other components required sentence analysis by the preprocessor and part-of-speech tagging by OTB. AutoSlog required additional processing associated with the key templates as well as OTB tags and sentence analysis with CIRCUS. The appositive component required OTB tags and NP analysis. TTG required a complete analysis of the entire development corpus by CIRCUS and all of the other trainable components. In general, there are many component dependencies in the system development cycle. If one of our components is updated or retrained, it is usually a good idea to retrain all the trainable components that depended on it during their training. This complicates the development cycle and adds time to what might be an otherwise "ideal" development cycle.

Manual programming is also needed to set up training data in various formats for each component. In some cases this amounts to little more than reading in all the files in a directory. In other cases, data collection demands a more ambitious programming effort (e.g. AutoSlog needs code that can relate template fills back to source texts). We have not recorded the time required to bring up the code needed for all the data collection, but most of this code can be readily recycled for new domains (AutoSlog is most likely to require special adjustments to handle new template specifications).

The amount of runtime needed for actual data collection prior to interactive training varies from component to component as does the amount of time needed for automated machine learning after interactive training. As a general rule, the amount of runtime for data collection and machine learning will be less than the amount of time needed for interactive training. So in an ideal system development cycle, we can just double the time spent on interactive training to get an estimate on the total amount of time needed to create a trainable component. (74 hrs on EJV -> 148 hrs). In the case of TTG where there is no interactive training, we estimate 40 hours for manual coding (setting up domain objects according to the domain guidelines) and 20 hours for data collection and 20 hours for the machine learning algorithm. TTG is our most intensive learning algorithm. (148 hrs -> 228 hrs)

As a rule, system development never proceeds in an ideal fashion. There are usually some false starts associated with data collection, and it may be necessary to run a machine learning algorithm more than once. Taking into account the time spent doing the same things more than once probably doubles the total amount of time estimated for an ideal system development cycle. (228 hrs -> 456 hrs). This suggests that a complete information extraction system for a new domain could be implemented in the space of 12 person/weeks, dividing that time about equally between a Lisp programmer who is familiar with our system development strategy and a domain expert who can operate the training interfaces. No expertise in natural language processing per se is required. Note that this time estimate does not include the time required to generate the key templates used by AutoSlog and TTG. Although key templates are labor intensive, they require only domain expertise.

We note that the actual time spent on our EJV and EME systems was much higher than the estimate given above because we were designing and implementing new code for many components, we went through more than one development cycle for each system component, we were experimenting with many variations on individual components during system development, and we were conducting many tests in conjunction with our experiments.

Some Caveats

The above time estimate for system development in a new domain assumes that there are many explicit connections between the source texts and the key templates. Although this assumption held for EJV, it was less valid for EME. Many of the template fills for EME were not string fills and much of the text processing needed for EME relied on extensive keyword recognition. AutoSlog was not originally designed to help us automate a CN dictionary based on keywords, so some analogous mechanism is needed to speed dictionary construction for keywords. This may be an easier problem than the dictionary construction problem that AutoSlog does handle, but we have not yet been able to evaluate AutoSlog's effectiveness on slot fillers that rely primarily on extensive synonym recognition.

Our time estimate also assumes that no new preprocessing specialists are needed. EJV and EME both use 5 specialists to recognize dates, locations, currency objects, percentages, and "aka" descriptions. These specialists are hand-coded and may or may not require adjustments if moved to a new domain. A new domain might benefit from different or additional preprocessing specialists. We estimate 40 hours of additional programming for each new specialist on average.

There is also a certain amount of "template massaging" that is needed to extract relevant information from noun phrases inside string fills. For example, a string fill for a person ID might contain a title that needs to be isolated and placed in a different slot. This optimization of template instantiations requires a manual coding effort that is difficult to estimate. As much as another 40 hours for response template manipulations may be needed to achieve a reasonable mapping into the more detailed template specifications.

Given these caveats, it appears that one programmer and one domain expert should be able to bring up a complete information extraction system in the space of two months. Such a system would undoubtedly benefit from additional analysis and modifications, depending on the specific needs of the intended user community.

Publications and presentations

April 24 site visit (Steve Dennis, Tom Crystal, Rita McCardell Doerr)

Cardie, C. (1993). "Using Decision Trees to Improve Case-Based Learning". in *Proceedings of the Tenth International Conference of Machine Learning"*. Amherst, MA. pp. 25-32.

Cardie, C. (1993). "A Case-Based Approach to Knowledge Acquisition for Domain-Specific Sentence Analysis". To appear in the *Proceedings of the Eleventh National Conference on Artificial Intelligence* (AAAI-93). Washington DC.

Riloff, E. "Automatically Constructing a Dictionary for Information Extraction Tasks". To appear in *Proceedings of the Eleventh Annual Conference on Artificial Intelligence. 1993*. Washington DC.

Lehnert, W.G. "Cognition, Computers and Car Bombs: How Yale Prepared Me for the 90's". To appear in *Belief, Reasoning, and Decision Making: Psycho-logic in Honor of Bob Abelson* (eds: Schank & Langer), Lawrence Erlbaum Associates. (in press)

Riloff, E. (1993) "Using Cases to Represent Context for Text Classification". To appear in *Proceedings of the Second International Conference on Information and Knowledge Management* (CIKM-93).

Riloff, E. and Lehnert, W.G. "Information Extraction as a Basis for High-Precision Text Classification," Submitted to *ACM Transactions on Information Systems* (Special Issue on Text Categorization). 1993.

Cowie, J. and Lehnert, W. "Information Extraction," Submitted to a special issue of the *CACM*. 1993.

Lehnert, W., Cardie, C., Fisher, D., McCarthy, J., Riloff, E. and Soderland, S. "Evaluating an Information Extraction System," Accepted pending revisions to the *Journal of Integrated Computer-Aided Engineering*. 1993.

Meetings attended

TIPSTER II planning meeting (Claire Cardie & Charles Dolan)

Schenectady, NY April 12-14

TIPSTER II planning meeting (Wendy Lehnert & Charles

Dolan) Cambridge, MA May 6-7

NLM Board of Scientific Counselors (Wendy Lehnert)

Bethesda, MD May 13-14

ARPA ISAT Study Group (Wendy Lehnert) Arlington, VA May 25-26

ARPA ISAT Study Group (Wendy Lehnert) Pittsburgh, PA June 16

MACHINE LEARNING CONFERENCE (Claire Cardie & Joe McCarthy)

Amherst, MA June 27-29

	training base	training time*	hit rates	ML techniques	manual code?
OTB (p-o-s tags)	1009 sentences	16 hours	97%	statistical	tag repair
MayTag (semantic tags)	174 sentences	14 hours	75%	CBR/d-trees	tag repair
NP analysis (termination)	1350 instances	14 hours	87%	d-trees	none
appositives (co-reference)	2276 instances	10 hours	87%	d-trees	none
AutoSlog (CN definitions)	1100 templates	20 hours	N/A	one-shot	none
TTG (template generation)	1100 templates	none	N/A	d-trees	domain objects

Table 1: EJV Trainable System Development

	training base	training time*	hit rates	ML techniques	manual code?
OTB (p-o-s tags)	621 sentences	10 hours	97%	statistical	tag repair
MayTag (semantic tags)	N/A	N/A	N/A	N/A	N/A
NP analysis (termination)	same as EJV	N/A	86%	d-trees	none
appositives (co-reference)	same as EJV	N/A	untested	d-trees	none
AutoSlog (CN definitions)	1000 templates	17 hours	N/A	one-shot	none
TTG (template generation)	1000 templates	none	N/A	d-trees	domain objects

Table 2: EME Trainable System Development

* this represents only the interactive time spent by a human in the loop

All trainable components were used for both EJV and EME with the exception of MayTag which was only used for EJV.

Conference Report

Title:	Corpus-Based Knowledge Acquisition Support for Text Analysis Systems
Frequency:	Quarterly Report
Period:	October 1, 1992 - December 31, 1992
Contract No.:	MDA904-92-C-2390
Submitted by:	Wendy G. Lehnert, Principal Investigator Department of Computer Science University of Massachusetts, Amherst, MA 01002
Security Classification:	Unclassified

Distribution Statement A: Approved for Public Release; distribution is unlimited

Conference Reports

(1) TIPSTER Phase II planning meeting (#1).

Date, place of conference, and attendees: April 12-14, 1993, Schenectady, NY, Charlie Dolan & Claire Cardie.

Subjects discussed: sharable NL resources, IE system architectures, system development.

(2) TIPSTER Phase II planning meeting (#2).

Date, place of conference, and attendees: May 6-7, 1993, Cambridge, MA, Charlie Dolan & Wendy Lehnert.

Subjects discussed: government application scenarios, recommendations for phase II.

(3) ARPA ISAT Study Group.

Date, place of conference, and attendees: May 25-26, Arlington, VA, Wendy Lehnert.

Subjects discussed: status reports for this year's summer study

(4) ARPA ISAT Study Group.

Date, place of conference, and attendees: June 16, 1993, Pittsburgh, PA, Wendy Lehnert.

Subjects discussed: focus meeting on the NII and higher education.

(5) Machine Learning Conference.

Date, place of conference, and attendees: June 27-29, Amherst, MA, Claire Cardie & Joseph McCarthy

Subjects Discussed: Claire Cardie presented a paper on her work. She followed up with discussions on ideas for improving the performance of the MayTag semantic feature tagger. The conference was held at UMass, so we were able to demonstrate two of our trainable language processing components at an AI Open House during

STATUS REPORT

Title: Corpus-Based Knowledge Acquisition Support for Text Analysis Systems

Frequency: Quarterly Report

Period: July 1, 1993 - December 30, 1993

Contract No.: MDA904-92-C-2390

Submitted by: Wendy G. Lehnert, Principal Investigator
Department of Computer Science
University of Massachusetts, Amherst, MA 01002

Security Classification: Unclassified

Distribution Statement A: Approved for Public Release; distribution is unlimited

UMass/Hughes TIPSTER Quarterly Report: July 1, 1993 - Dec. 30, 1993

Tipster 4th Quarter Report

Our activities during this period were dominated by the completion of the 24 month evaluation at the end of July. subsequent data analysis and report generation for the MUC-5 and Tipster 24-month meetings. We were also able to complete a dictionary construction experimental with government analysts in time for presentation at the 24-month meeting in September.

System Development Work

Final preparations for the 24-month evaluation addressed trainable template generation and trainable coreference components. We really needed more time to develop these components, but we did not establish upstream system stability until the end of June, and this made it impossible to collect useful training data before July. in retrospect, it is clear that we underestimated the amount of time needed to develop a large number of trainable system components operating in a serial architecture: work on the final components could not begin until all preceding components were relatively stable. This left us with far too little time to work on template generation and coreference. Nine months was just not enough time for a full system development cycle in two domains for Tipster Phase I.

In the end, we failed to incorporate trainable coreference in the evaluation system, and our last-minute effort to bring up a heuristic coreference module was minimally adequate. We do not feel that our ideas on trainable coreference had a fair trial at the 24-month evaluation because of the unrealistic development schedule.

Lessons from the Evaluation

Our evaluation results have been reported in detail in the 24-month evaluation proceedings. Because we were working with trainable components, we were able to assess the performance of our individual system components in addition to the overall system evaluation. This allowed us to present something closer to a "glass box" evaluation report in addition to the shared "black box" evaluation. At the component level, we were very pleased with the results of our OTB tagger, AutoSlog dictionary construction, MayTag semantic tagger, noun phrase terminator, and CIRCUS sentence analyzer. Coreference and template generation appeared to be our weakest links for the reasons explained above. Since these last two components performed critical functions, our overall system evaluation was not as strong as we would have liked.

We also note that our internal test runs on EME were significantly higher than those obtained in the official evaluation, because we tuned our coreference heuristics for EME on the basis of test runs using the EME tested from the 18-month evaluation. Unfortunately, this test set turned out to not be very representative of the EME corpus in general since a disproportionate number of the texts in this test set were relatively short. The details of this discrepancy are presented in our 24-month summary report.

A Dictionary Construction Experiment

In August we were pleased to conduct a 2-day experiment with two government analysts in order to test our claims about automated dictionary construction. Our experiment confirmed that domain knowledge is more important than linguistic expertise when it comes to creating a domain-specific dictionary using the AutoSlog dictionary construction tool. Although the 48-hour time frame prohibited us from obtaining complete EJV dictionaries from our subjects, we were able to scale back the size of the dictionaries under construction, and compare the performance of the experimental dictionaries against an analogously scaled-back version of our official 24-month dictionary. Our experiment showed that the dictionaries constructed by the government analysts were largely indistinguishable from the dictionary constructed by researchers at UMass.

We have prepared a special report for the 24-month Tipster proceedings detailing the design and results of this experiment.

Publications:

Riloff, E. and Lehnert, W.G. "A Dictionary Construction Experiment with Domain Experts", in *Proceedings of the Final Tipster Evaluation Meeting* (in press).

Lehnert , W., J. McCarthy, S. Soderland, E. Riloff, C. Cardie, J. Peterson, F. Feng, C. Dolan and S. Goldman, "University of Massachusetts/Hughes: Description of the CIRCUS System Used for TIPSTER Text Extraction" in *Proceedings of the TIPSTER Text Phase I 24-Month Conference, Executive Review*. pp. 69-71.

Lehnert, W., J. McCarthy, S. Soderland, E. Riloff, C. Cardie, J. Peterson, F. Feng, C. Dolan and S. Goldman, "University of Massachusetts/Hughes: Description of the CIRCUS System as Used for MUC-5," to appear in *Proceedings of the Fifth Message Understanding Conference.*

Cardie, C. (1993) "A Case-Based Approach to Knowledge Acquisition for Domain-Specific Sentence Analysis" in *Proceedings of the Eleventh National Conference for Artificial Intelligence* (AAAI-93) pp. 798-803.

Riloff, E. "Automatically Constructing a Dictionary for Information Extraction Tasks". in *Proceedings of the Eleventh Annual Conference on Artificial Intelligence. 1993.* pp. 811-816.

Riloff, E. (1993). "Using Cases to Represent Context for Text Classification" in *Proceedings of the Second International Conference on Information and Knowledge Management* (CIKM-93) pp. 105-113.

Conference Report

Title:	Corpus-Based Knowledge Acquisition Support for Text Analysis Systems
Frequency:	Quarterly Report
Period:	July 1, 1993 - Dec. 30, 1993
Contract No.:	MDA904-92-C-2390
Submitted by:	Wendy G. Lehnert, Principal Investigator Department of Computer Science University of Massachusetts, Amherst, MA 01002
Security Classification:	Unclassified

Distribution Statement A: Approved for Public Release; distribution is unlimited

Conference Report

(1) American Association for Artificial Intelligence

Date, place of conference and attendees: July 10-16, 1993, Washington DC. Wendy Lehnert, Claire Cardie, and Ellen Riloff.

Subjects discussed: Claire and Ellen each presented papers on progress in information extraction technologies, and Wendy was there as a program co-chair for the conference. This was the first year that AAAI presented a technical session on information extraction. Professor Lehnert also arranged an invited talk by Eugene Charniak on statistical NLP methods.

(2) NSA

Date, place of conference and attendees: August 18-21, 1993, Fort Mead, MD. Ellen Riloff

Subjects discussed: The purpose of this trip was to conduct a dictionary construction experiment with government analysts. The first day was devoted to settingup system software on two MAC computers at Fort Meade. Days two and three were devoted to the actual experiment.

(3) ARPA ISAT meeting

Date, place of conference and attendees: August 17-24, 1993, Woodshold, MA. Wendy Lehnert

Subjects discussed: As a member of the standing ISAT committee, Professor Lehnert participated in the subgroup on education. At the close of this meeting, a final presentation of the 1993 ISAT Study Group on technical requirements for the "Information Highway".

(4) Fifth Message Understanding Conference

Date, place of conference, and attendees: August 24-27, 1993, Baltimore, MD. Joe McCarthy, Jon Peterson, Stephen Soderland, Wendy Lehnert, Charlie Dolan, and Seth Goldman.

Subjects discussed: Presentations were made by Wendy, Joe, and Charlie. Joe gave system demos. We had many useful technical interactions with other system developers, and our system demos attracted a number of government attendees.

(5) Tipster 24-month meeting

Date, place of conference, and attendees: Sept. 19-23, 1993, Fredericksburg, VA. Joe McCarthy, Wendy Lehnert, Ellen Riloff and Charlie Dolan

Subjects discussed: Presentations were made by Wendy and Charlie. Joe and Stephen gave system demos. We were able to report on the dictionary construction experiment at this meeting in addition to the results of the 24-month evaluation.